arXiv:1501.05382v2 [cs.CV] 28 Jan 2015

Enhanced Mixtures of Part Model for Human Pose Estimation

Wenjuan Gong, Yongzhen Huang, Jordi Gonzàlez, and Liang Wang

***Abstract*—Mixture of parts model has been successfully applied to 2D human pose estimation problem either as explicitly trained body part model or as latent variables for the whole human body model. Mixture of parts model usually utilize tree structure for representing relations between body parts. Tree structures facilitate training and referencing of the model but could not deal with double counting problems, which hinder its applications in 3D pose estimation. While most of work targeted to solve these problems tend to modify the tree models or the optimization target. We incorporate other cues from input features. For example, in surveillance environments, human silhouettes can be extracted relative easily although not flawlessly. In this condition, we can combine extracted human blobs with histogram of gradient feature, which is commonly used in mixture of parts model for training body part templates. The method can be easily extend to other candidate features under our generalized framework. We show 2D body part detection results on a public available dataset: HumanEva dataset. Furthermore, a 2D to 3D pose estimator is trained with Gaussian process regression model and 2D body part detections from the proposed method is fed to the estimator, thus 3D poses are predictable given new 2D body part detections. We also show results of 3D pose estimation on HumanEva dataset.**

***Index Terms*—Pose estimation, double counting problem, mixture of parts Model**

I. Introduction

Pose estimation from still images has wide applications in image and video indexing, video surveillance and human computer interaction. For example, online solutions of this problem can be applied for single frame initialization in tracking human poses. Yet pose estimation from still images, that is, 2D body part localization is a difficult problem, due to the fact that human body is highly flexible resulting human poses with high degrees of freedom even in 2D images.

A state-of-art and currently widely used solution for 2D body part detection is the mixture-of-parts (MoP) method [1], in which a human body is modeled as a tree structure and body parts are encoded as nodes in the tree. Maximum responses from detection are passed from the leaf nodes to the root. One problem with this solution is the double-counting problem, that is, one detected body part is counted twice for both sides of the human body. In this paper, we tackle the double-counting problem in MoP model with multiple feature inputs. Additional input features are incorporated so we are able to verify body part localization from more feature responses. Compared with the greedy solution in the original MoP method, we are able to solve double-counting problem with a global optimization.

With detected 2D body part locations from the proposed method, we are able to predict 3D poses by feeding 2D body part detections to a 2D to 3D pose estimator. For 2D to 3D pose estimator, we choose Gaussian process regression, which has been proved to be effective in modeling non-linear regression problems. We further validate the whole pipeline on a public available dataset for pose estimation: HumanEva dataset. We visualized two types of results: enhanced 2D body part detections and 3D poses estimated from enhanced 2D body part detections. Figure 1 shows main steps as a pipeline for the whole algorithm. The pipeline includes several major steps: feature extractions (MoP and background subtraction, in our case), global optimization, and 2D to 3D pose estimation. From the input, the original mixtures of part model is trained and applied to detect body part positions. Meanwhile, human blobs extracted with background subtractions are used as another cue in our method. Then in the third step, this two cues are combined with the proposed algorithm.

The generalized framework in our algorithm are able to incorporate multiple features other than human blobs. We extract multiple cues from input images and combine them under the proposed framework. With augmented inputs from multiple features, we are able improve 2D body part localization and solve double counting problem. As in MoP method, a human body is modeled by a tree structure where kinematic constraints between connecting parts are kept. First, feature models of different features are trained separately and the optimization target is modified to reach a global optimization target by incorporating multiple feature cues. The inference of the optimal pose in a test image is also carried out with multiple cues. The bottom up message passing procedure combines multiple feature cues so as to reduce false positive body part detections and the top down back tracing procedure are optimized globally so as to tackle double counting problem.

The contributions of this papers are as followings: by combing multiple cues, we boost 2D body part localizations under a general framework; enhanced 2D body part detectors are validated on a public available dataset: HumanEva dataset; 3D pose estimation is shown as an examplar application of detected 2D poses and this application is also validated on HumanEva dataset. The rest of the paper is organized as following: in section II, we introduce related works on 2D pose estimation and related works on solving double counting problems; in section III, we introduce details of

W. Gong is with the Department of Computer and Communication Engineering, China University of Petroleum, Qingdao, China, e-mail: wenjuangong@upc.edu.cn.

Y. Huang and W. Liang are with Institute of Automation, Chinese Academy of Science.

J. Gonzàlez is with Computer Vision Center, Autonomous University of Barcelona.

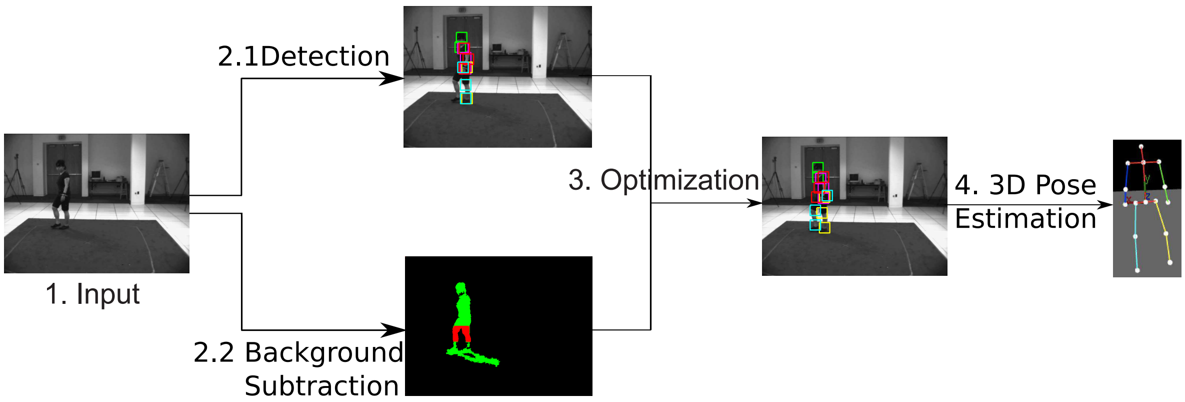

Fig. 1. 2D body part detections and 3D pose estimation based on enhanced Mixtures of Parts method. The pipeline includes several major steps: feature extractions (MoP and background subtraction, in our case), global optimization, and 2D to 3D pose estimation.

the proposed method including training models from different features and the global optimization taget; section IV shows boosted 2D body part localizations by combining multiple feature cues on a standard public available dataset and the examplar application of 3D pose estimation from localized 2D body parts; in section V we conclude the work and discuss about possible future works.

II. Related Work

As mentioned in the previous section, human bodies are highly flexible, thus results in a huge amount of possible guesses in the solution space. But human body joints are not completely under no constraints. Models like pictorial structure model [2] and tree models [3], [4], [1] are exploited and successfully applied to represent human body models in 2D. These models keep kinematic constraints between connecting body parts. That is, the body parts that are connected physically are also connected in the tree structure. Using tree structures has the advantage of tractable inference of the optimal pose. However, spatial constraints between body parts without direct connections are not incorporated in the tree structure. Due to this reason, the original tree structure cannot deal with occlusion and has the problem of double counting, where an image evidence is counted twice for different body parts.

One important example of tree models is mixture of parts model [1]. MoP defines body parts as an area surrounding body joints and has the advantage in dealing with the foreshortening problem of body limbs caused by viewpoint changes. While the traditional pictorial structure (PS) model [2], which defines body parts as body limbs, needs to deal with foreshortening problem by explicitly training on body limbs of different lengths. Also in MoP model, the orientation of a limb is naturally represented by the connection of detected body joints. While in a traditional PS model, limb orientations need to be learned and detected explicitly. So we choose MoP model as the human body model. A body part in the MoP model is represented as a mixture of several templates, each of which is trained with one subset of samples of this body part. In this way, the trained body part is able to deal with different limb layouts from different poses.

As mentioned in the first paragraph, although the tree structured human model is efficient in training and referencing, it has the double counting problem due to occlusions and lack of constraints between body parts denoted as a node in the tree structure. To deal with these problems, authors in [4] propose multiple tree models. The models contain a tree structure to account for kinematic constraints between connected body parts, tree structures for spatial constraints among body parts without direct connections, and tree structures for occluded body parts. Different tree structures are combined with a boosting procedure. Other research also explore the possibility of imposing constraints in the optimization target. For example, authors in [3] modify the optimization target and incorporate spatial constraints to deal with double counting problem. In referencing, those poses who violate the spatial constraints will get a comparatively lower score.

III. The Method

Given training images with only one human in each image, we train 2D body part detectors with image patches cropped within bounding boxes surrounding the body parts. For a test image, we localize 2D body part positions with trained detector and optimize the detection with multiple feature cues. We name the detector enhanced MoP model since it is based on the MoP model proposed in [1]. In the following subsection, we are going to split the method into modules and explain in details.

A. *Mixture of Parts*

The idea of mixture of parts detector in [1] is to represent a body part with a mixture of several (5 or 6) templates, each represent a different appearance of the corresponding body part. So the body part which has more variances in appearance, for example, elbows and knees, are apt to contain more templates. After cropping the image surrounding the bounding box with a proper size, all samples of a body parts are clustered into several clusters, whose total number is predefined according to the variance of the body part. Training templates are formulated as optimizing parameters in a support vector machine, which is carried out with EQ optimization.

Note that the size of the bounding box is a crucial factor in adapting the method to custom data. Considering the different notation in each dataset, body joints might correspond to different position and if the size of the bounding box is defined too big, it might contain information from other body part and if the size is too small, it might be lack of information for identifying the body part or joint.

After training templates for each body part, given a test image, we compute the response of the image with respect to all trained templates by convolution. Then a distance transform [5] is performed so that the maximum response of the image to the test template is highlighted. Later on, we start from the leaves of the human tree structure (rooted at the head), and pass maximum responses of all mixtures from the child body part to its parent. Thus, when we come to the root node, all the body part nodes contribute by passing messages. The score of the root is considered the final score of the human detection. This tree structure is very effective in referencing but it has problem dealing with double counting problem. In the following subsections, we are going to explain how we are going to enhance the algorithm based on the original MoP model.

B. Enhanced MoP Via Multiple Cues Fusion

Instead of imposing spatial constraints or modifying tree structure model, we explore the possibility of combining multiple cues from input images. We argue that multiple feature cues provides richer information so that effectively combining multiple cues reduce false positives and ease double counting problem. In our experiments, we consider histogram of gradient (HOG) [6] and human blobs extracted from background subtraction [7].

1) Formulation of Enhanced Model: Let us write I for an image, $p_i = (x\ y)$ for the pixel location of part i and t_i for the mixture component of part i. We write $i \in \{1, \ldots, K\}$, $p_i \in \{1, \ldots, L\}$ and $t_i \in \{1, \ldots, T\}$. We call t_i the "type" of part i. For notational convenience, we define the lack of subscript to indicate a set spanned by that subscript (e.g., $t = \{t_1, \ldots, t_K\}$). The kinematic constraints of human body between connected body parts are modeled as following:

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j}. \tag{1}$$

The parameter $b_i^{t_i}$ favors particular type assignments for part i, while the pairwise parameter $b_{ij}^{t_i,t_j}$ favors particular co-occurrences of part types. We write $G = (V; E)$ for a K-node relational graph whose edges specify which pairs of parts are constrained to have consistent relations.

We can now write the full score associated with a configuration of part types and positions:

$$S(t) = S(t) + \sum_{i \in V} \omega_i^{t_i} \cdot \phi(I, p_i) + \sum_{ij \in E} \omega_{ij}^{t_i, t_j} \cdot \psi(p_i - p_j), \tag{2}$$

where $\phi(I, p_i)$ is a HoG vector extracted from pixel location p_i in image I. $\psi(p_i - p_j) = [dx \quad dx^2 \quad dy \quad dy^2]^T$, where $dx = x_i - x_j$ and $dy = y_i - y_j$, the relative location of part i with respect to j.

Until now, this is the original MoP model from [1]. Since multiple features are extracted separately, we can train models from each candidate features separately. For example, when we use extracted human blobs as another feature cue. We get the human blob model from background subtraction as following:

$$F(p_i) = \begin{cases} 1, if|E(p_i) - B(p_i)| > threshold, \\ 0, if|E(p_i) - B(p_i)| < threshold, \end{cases} \tag{3}$$

where $B(p_i)$ is the background model and can be updated with new added frames in the following way,

$$B_{t+1}(p_i) = \alpha * F_t(p_i) + (1 - \alpha) * B_t(p_i). \tag{4}$$

And α is the learning rate. We denote this model as human blob (HB) model.

After training MoP model and HB model separately, given a test image, we need to find the optimal human pose with respect to certain criterion. This criterion should take into account both of the trained models. Since we suppose each image features are extracted separately. We can get the joint probability of matching two models as:

$$P(M, H) = P(M|H) \cdot P(H), \tag{5}$$

where M represents the MoP model and H represent the HB model. This probabilities formulation can be easily extended to other image feature cues, given the definition of the model probabilities and conditional probabilities.

In our method, we consider HOG and detected human blobs. We define $P(H)$ for each pixel m as following:

$$P_m(H) = \begin{cases} 1, & ifF_m = 1, \\ o, & ifF_m = 0. \end{cases} \tag{6}$$

Since the HOG feature and the human blob feature are extracted separately and thereafter MoP model and HB model are trained separately, $P(M|H)$ equals $P(M)$ which is defined in equation (2). In implementation, we calculate the probability of a certain pixel m belonging to a certain body part by convoluting with image evidence of this pixel with trained body part template.

2) Finding Root Positions: After training MoP model and HB model separately from HOG and human blob features. We can detect human pose from an unseen image by find the optimal human pose. In [1], with all trained mixtures of parts models, the test image is convoluted with each trained templates. Then starting from the leaves of the tree structure, responses of all body parts are passed to their parent parts. After one pass, all the body parts contribute their score to the root part of the tree structure.

In our combined model, before passing the score from the child node to its parent node, we check if this pixel also confirms with the evidence from human blob detection. If the current pixel belongs to the detected human blob, we keep the current score, otherwise the score is set to a very small value. This procedure guarantee that the final probability is the joint probability of two candidate feature models. The advantage of this procedure is obvious, we can remove some false positives by verifying that the current pixels confirm with both models trained from different image feature cues. So the detected root position is more accurate. After we find the root position of the human, we go through the whole tree to fix each body parts with global optimization.

3) Finding Body Part Positions: From the detected root position of the tree structure, authors in [1] employ a backtracking algorithm to fix all the body part positions. It starts from the root of the tree structure and fix its child node by picking the maximum response from all the child nodes. This method causes the double counting problem. Since each body part is fixed only considering the response of the test image with the trained templates, when sibling body parts (the same body part, but on different sides of the human body, like a left hand and a right hand) resemble each other, the same image patch might be picked repetitively. In this case, the estimated pose is occluded while in the real case it is not.

To solve this problem, we use global optimization to fix each body part positions. In the original MoP model, where there is only HOG feature, optimizing body part position is very time consuming. For example, if the model uses 26 body parts, and each body parts use 5 or 6 mixtures, the minimum number of possible combination for all body parts is 5^25. This is a huge amount of possible guesses. In our case, where we consider human blob as another feature cues. The possible number of mixtures for each body parts is great reduced due to the constraint. So we can optimize the tree structure globally. The body part positions are optimized to maximize the score of the proposed model for combining multiple cues S_{MC}:

$$\sum_{i \in V} \omega_i^{t_i} \cdot \phi(I, p_i) \cdot F(p_i) + \sum_{ij \in E} \omega_{ij}^{t_i, t_j} \cdot \psi(p_i - p_j) \cdot F(p_i) \cdot F(p_j), \quad (7)$$

where $F(p_i)$ is the ratio of overlap between the body part p_i and the foreground model define in equation(3).

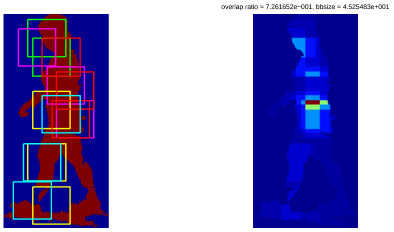

Fig. 2. Detected body part localization with the Mixture of Parts (MoP) model. The left figure shows the localizations of all body parts with MoP. The right figure shows the overlap between extracted human blobs and all bounding boxes. Color variations in human blobs denotes different number of bounding boxes that are overlapped.

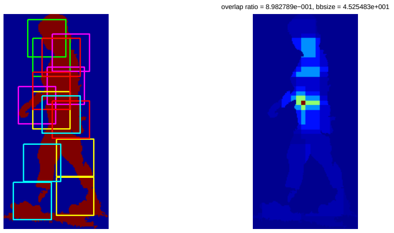

Fig. 3. Body part localization after optimization (left) and final overlap map between all bounding boxes and the extracted human blob.

C. From 2D Parts to 3D Pose Estimation

The Gaussian process regressor is one of the most widely used regression model for learning 2D to 3D mapping in the pose estimation since it has been proved to be an effective approach for the nonlinear 2D to 3D pose mapping problem [8], [9], [10]. Gaussian Process Regression (GPR) is considered a model-free framework. GPR defined as a distribution over functions, extends statistics from data points to functions. With kernel trick, we can even get rid of the function definition, and only concentrate on kernel matrix instead. Once we normalize the training input to have a zero mean, we only need to define a covariance matrix, that is ,the kernel matrix, for GPR. Frequently used covariance matrices include squared exponential covariance function, Matérn covariance function and so on. In the following subsections, we will explain detailed representations and settings for the Gaussian process regressor used here.

1) Definition of Gaussian Process Regression: According to [11], Gaussian process is defined as: *a collection of random variables, any finite number of which have (consistent) joint Gaussian distribution*. A Gaussian process is completely specified by its mean function and a covariance function. Integrating with our problem, we denote the mean function as $m(\mathbf{s})$ and the covariance function as $k(\mathbf{s}, \mathbf{s}')$, so a Gaussian process is represented as:

$$\zeta(\mathbf{s}) \sim \mathcal{GP}_j(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')), \quad (8)$$

where

$$\begin{aligned} m(\mathbf{s}) &= E[\zeta(\mathbf{s})], \\ k(\mathbf{s}, \mathbf{s}') &= E[(\zeta(\mathbf{s}) - m(\mathbf{s}))(\zeta(\mathbf{s}') - m(\mathbf{s}'))], \quad (9) \end{aligned}$$

2) Hyperparameter Optimization and Referencing: We assume prediction noise as a Gaussian distribution and formulate finding the optimal hyperparameters as an optimization problem. We seek the optimal solution of hyperparameters by maximizing the log marginal likelihood (see [11] for details):

$$\log p(\Psi' | \mathbf{s}, \theta) = -\frac{1}{2} \Psi'^T K_{\Psi'}^{-1} \Psi' - \frac{1}{2} \log |K_{\Psi'}| - \frac{n}{2} \log 2\pi, \quad (10)$$

where $K_{\Psi'}$ is the calculated covariance matrix of the target vector.

With the optimal hyperparameters, the prediction distribution is represented as:

$$\begin{aligned} \Psi'^* | \mathbf{s}^*, \mathbf{s}, \Psi' \sim \mathcal{N}(\mathbf{k}(s^*, \mathbf{s})^T [K + \sigma_{noise}^2 I]^{-1} \Psi', \\ k(s^*, s^*) + \sigma_{noise}^2 - \mathbf{k}(s^*, \mathbf{s})^T [K + \sigma_{noise}^2 I]^{-1} \mathbf{k}(s^*, \mathbf{s})), \quad (11) \end{aligned}$$

where K is the calculated covariance matrix from training 2D image features s and σ_{noise} is the covariance of Gaussian noise.

Equation11 for referencing test data is deducted from marginal and conditional properties of Gaussian distributions. The following is the marginal property of Gaussian distributions: the marginal of a joint Gaussian is again a Gaussian, that is,

$$\begin{aligned} & p(\mathbf{x}, \mathbf{y}) = \mathcal{N}(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}) \\ \Rightarrow \quad & p(\mathbf{x}) = \mathcal{N}(\mathbf{a}, A). \quad (12) \end{aligned}$$

And the conditional property of Gaussian distributions are: the conditionals of a joint Gaussian are again Gaussian, that is,

$$\begin{aligned} p(\mathbf{x},\mathbf{y}) &= \mathcal{N}(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}) \\ \Rightarrow \quad p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{a}+BC^{-1}(\mathbf{y}-\mathbf{b}), A-BC^{-1}B^T). \quad (13) \end{aligned}$$

Thus we are able to predict the distribution of x given the distribution y.

In most cases, we assume that Gaussian process priors have zero means, that is,

$$f(x)|M_i \sim \mathcal{GP}_j(m(\mathbf{x}) \equiv 0, k(\mathbf{s},\mathbf{s}')). \quad (14)$$

This leads to a Gaussian process posterior

$$f(x)|\mathbf{x},\mathbf{y},M_i \sim \mathcal{GP}_j(m_{post}(\mathbf{x}), k_{post}(\mathbf{s},\mathbf{s}')), \quad (15)$$

where

$$m_{post}(\mathbf{x}) = k(x,\mathbf{x})[K(x,\mathbf{x}) + \sigma^2_{noise}I]^{-1}\mathbf{y}. \quad (16)$$

With this posterior, we only need to define covariance matrices, known as kernel in machine learning community.

The most frequently used covariance matrices (kernels) include: squared exponential (SE), Rational quadratic (RQ), Matérn and Periodic, smooth covariance functions. The function of covariance function is to define the distance measure in a newly transformed space where the original data samples have one to one correspondences with their mapped points and due to the transformation, data samples of different attribute classes in the new spaces are easier to classify or identify. With the kernel trick, we can get rid of directly defining the mapping model and only define the kernel matrix, the covariance matrix here.

3) GPR for 2D to 3D Pose Mapping: Gaussian processes yield a method for specifying a probability distribution over functions by specifying a mean and a covariance function for the function values $f(x)$. By training a Gaussian process with sample data $\{x, f(x)\}$ the variance of the Gaussian process becomes small for function values $f(x)$ at supporting points x included in the training data, which corresponds to an increased certainty about the function values at these points, while at other points x' the variance of the Gaussian process remains high which corresponds to a high uncertainty about the function values $f(x')$ at such points.

In our algorithm, we select the most commonly used covariance matrix: squared exponential covariance matrix. Given a 2D pose estimate which is represented as the 26 dimensional vector BP ($13 * 2$, where 13 is the number of body joints in MoP and 2 is the dimension size), we train one Gaussian process to predict each of the 60 dimensions of the 3D pose vector ψ ($20 * 3$, where 20 is the number of body joints in HumanEva motion capture data and 3 is the dimension size) separately. Then given features from test samples, we predict 3D poses with trained GPR.

Exp.	Action	Actor	TrFrmNo	TeFrmNo
1	Walking	$S1$	200	21
2	Walking	$S2$	200	21
3	Walking	$S3$	200	21
4	Box	$S1$	200	21
5	Box	$S2$	200	21
6	Box	$S3$	200	21

TABLE I
THE COMPOSITION OF EXPERIMENTS FROM HUMANEVA DATA SET. EXP. “TRFRMNO” DENOTES TOTAL FRAME NUMBERS FOR TRAINING. “TEFRMNO” DENOTES TOTAL FRAME NUMBERS FOR TEST.

IV. RESULTS

To demonstrate the effectiveness of the proposed method, we first show improved 2D body part localization results with the proposed feature fusion method; then, the trained 2D to 3D pose estimator is carried out on detected 2D body part locations and 3D poses are estimated and shown.

A. Evaluation Data and Experiment Settings

From HumanEva-I data set, we select two different actions (“Walking” and “Box”) performed by three different actors (“S1”, “S2” and “S3”). All three performers perform the actions within a fixed area (confined with a carpet). “Walking” is performed in a cyclic way, while in “Box”, performers are moving in a very small area positions notwithstanding different performing style.

As a result, we have six different experiments in total. Training and test are carried out within each experiment. That is, we train a detector on a single experiment setting and validate the trained models on the test frames of same experiment setting. This experiment setting is designed to compare the influence of different action type and different performer to the body part localization and pose estimation results. The detailed splits between training and test is shown in the following table.

For each action performed by a specific actor, training data are composed of 200 consecutive frames, which is close to the number of frames in a cyclic walking sequence. Test data are sampled with a equal step among the whole motion sequence excluding the training frames, so that the test poses covers all possible poses for an action.

B. Enhanced 2D Part Locations

The proposed 2D part localization method aims to solve double counting problem, that is, a pixel location (a body part location in our case) might be designated to two body limb positions even there is no occlusions between these body limbs. The reason for the double counting problem in MoP detection is:

1) the responses of a pixel location (or a body part observation) to all trained body part templates are calculated separately,
2) then, from leaf nodes to root nodes, a best response is selected for a each node among all candidate mixtures and this response is passed as a message to its parent node, that is, a locally optimal solution.

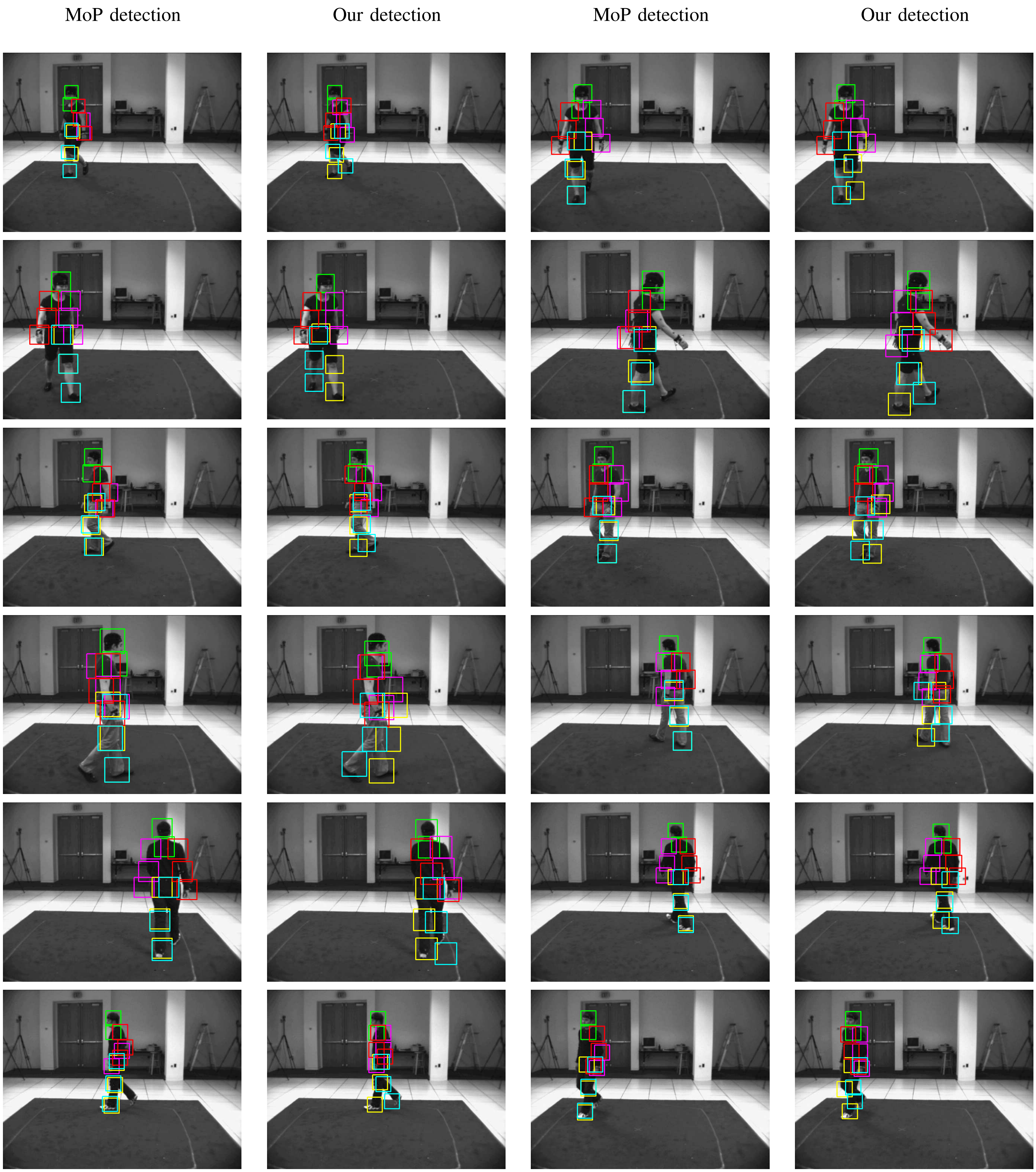

Fig. 4. Qualitative 2D pose estimation samples. Column "MoP detection" shows detected 2D body parts from the original MoP method. Column "Our detection" shows the enhanced 2D body part detection from our method.

The limitation of this solution is that usually, the chained body part position calculated from local optimums are not a global optimum and the essence of the solution gives no globally target.

In our solution, we introduce another feature cue (extracted human blobs in current experiments). In this way, not only the localization could be verified by two features, but also we are able to optimize the target in a global way. This is due to the newly introduced feature cue gives global description. The following are main module that are incorporated:

- we keep response scores from all mixtures of the current body part for later use. Instead of selecting the mixture with the maximum response in [1], we select pre-defined set of body part candidates whose overlap with another feature are over a certain ratio and put them in a candidate list, and calculate the best configuration whose overlap of two features are the maximum. The reason that we select a *pre-defined* set of body part candidates is that if we consider all body limbs, the calculation number might be too much ($5 * 5 * 5 * 6 * 6 * 5 * 6 * 6 * 5 * 6 * 6 * 5 * 6 * 6$ in our method, where we use fourteen body parts, with five or six mixture for each body part) and redundant because not all body limbs are possible to have double counting problem. The body parts that are possible to have double-counting problems are: two elbows (left and right), two hands, two knees and two legs. So we *pre-define* a set of body parts that could be optimized.
- we add a mixture-selection module, where any mixture that has a overlap ratio of over a certain threshold ($thresh2 = 0.2$ in our experiment, due to the noisy extracted silhouettes) are considered as a candidate mixture that can pass messages to it parent.
- we optimize the position among all kept candidate positions for a body part by fixing all other body part positions. If the overlap of two feature cues at a pixel location is within a interval ($thresh1 = 0.5$ in our experiment), for all pair of body part that might encounter double counting problem (that is, two elbows, including the left elbow and the right elbow, two hands, two knees and two legs), we check if they overlap. If they do, there is a possibility that they are double-counted, then they are added to the candidate list for local optimization.

Note that, in our experiments parameters are set by experience. It is also straightforward to acquire them from training data. For example, we can calculate all overlap ratios between bounding boxes of training body parts and extracted human blobs, fit Gaussian distribution, and take the mean of the fitted Gaussian as $thresh2$. Body part localization results are shown in figure 4. From the figure, we can see that double counted body parts are correctly localized after optimization.

C. 3D Pose Estimations

We further feed enhanced 2D body part locations to pose estimators and 3D poses are predicted. For each experiment, we train a set of Gaussian processes with Squared Exponential covariance matrix with the training set, the proposed 2D body part detectors are applied on test images, and detected 2D body parts are fed to the trained Gaussian process regressors to get 3D pose estimations. Here we show some visualizations of 3D pose estimations. Figure 5 shows examples from walking and box actions.

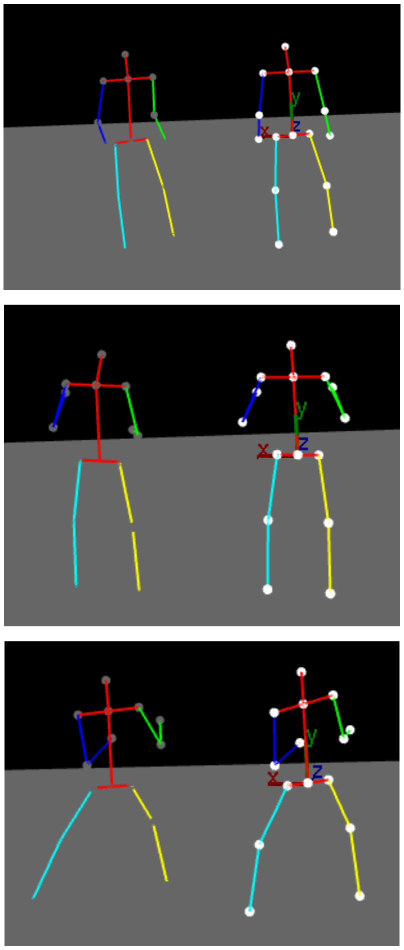

Fig. 5. Examples of visualized 3D pose estimation. The first row is a frame from actor S1 performing walking. The second row is a frame from actor S1 performing box. The third row is a frame from actor S2 performing box. The stick figure on the left is the ground truth data, and the stick figure on the right is the estimated 3D pose from localized 2D body part positions.

To have a qualitative comparison, we show in figure 6 3D joint positions of the ground truth pose and the estimated 3D pose. Both figures shows values from the first dimension of 3D joints. The figure in the first row is from the left elbow of the actor “S2” performing “Walking” and the figure in the second row is from the left hand of the actor “S2” performing “Walking”.

One direct application of the proposed method is for controlling 3D poses of avatar. Motion capture systems usually requires invasive body markers. While in our pipeline, performers are able to get rid of invasive body markers once training 3D poses are attained. What’s more, we only need image sequences from one single view point.

V. Conclusions and Discussions

In this paper, we design an algorithm to enhance the performance of 2D body part localization based on Mixture of Parts models which recently achieved good performances in 2D body part localization. Later on, we take the estimated poses as an input to estimate 3D poses. We validate our method in two ways: 2D body part localization visualized results and 3D pose estimation errors. One interesting further work is to incorporate physical constraints into 3D human model so we

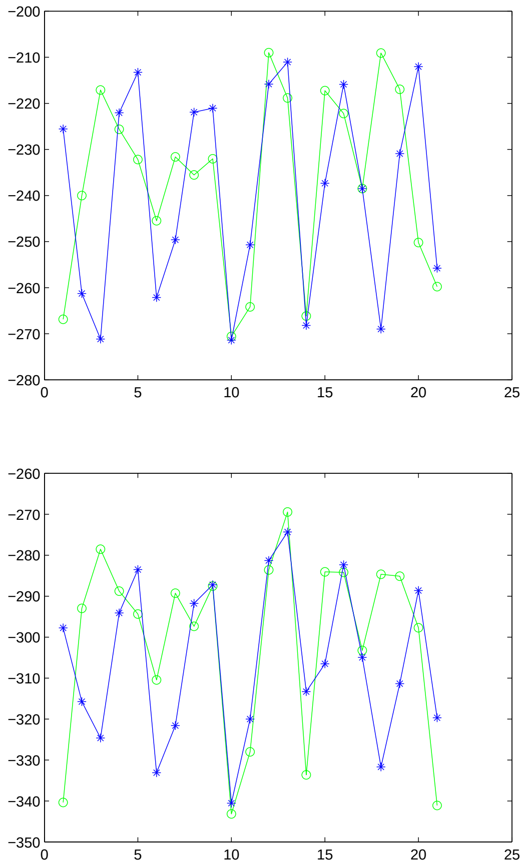

Fig. 6. The first dimension (values along axis x) of the left elbow position estimations (in green) and ground truth joint positions (in blue). The first figure is a frame from actor S2 performing walking. The second figure is a frame actor S2 performing walking. The x axis denotes frame id, ranging from 1 to 21. The y axis denotes values of the first dimension from 3D joint positions and its unit is milimeter.

can optimize 2D body parts accordingly. We are also interested into validate this method on other public data set, like YouTube data set where 3D pose ground truth are not provided.

References

[1] Y. Yang and D. Ramanan, *Articulated pose estimation with flexible mixtures-of-parts*, in Proceedings of Computer Vision and Pattern Recognition, 2011, pp. 1385-1392.

[2] P. F. Felzenszwalb and D. P. Huttenlocher, *Pictorial structures for object recognition*, Int. J. Comput. Vision, vol. 61, no. 1, pp. 55-79, Jan 2005.

[3] Y. Xiao, H. Lu, and S. Li, *Posterior constraints for double-counting problem in clustered pose estimation*, in IEEE International Conference on Image Processing, 2012, pp. 5-8.

[4] Y. Wang and G. Mori, *Multiple tree models for occlusion and spatial constraints in human pose estimation*, in Proceedings of European Conference on Computer Vision, 2008, pp. 710-724.

[5] P. F. Felzenszwalb and D. P. Huttenlocher, *Efficient graph-based image segmentation*, Int. J. Comput. Vision, vol. 59, no. 2, pp. 167-181, Sep 2004.

[6] N. Dalal and B. Triggs. *Histograms of oriented gradi- ents for human detection*. In CVPR, pages 886893, 2005.

[7] A. Amato, M. Mozerov, A. Bagdanov, and J. Gonzàlez, *Accurate moving cast shadow suppression based on local color constancy detection*, IEEE Transactions on Image Processing, vol. 20, pp. 2954-2966, 2011.

[8] Gregor Gregorčič and Gordon Lightbody. *Gaussian process approach for modelling of nonlinear systems*. Engineering Applications of Artificial Intelligence, 22(4-5):522533, 2009.

[9] B.-B. Sofiane and A. Bermak. *Gaussian process for nonstationary time series prediction*. Computational Statistics and Data Analysis, 47(4):705712, 2004.

[10] J. M. Wang, D. J. Fleet, and A. Hertzmann. *Gaussian process dynamical models for human motion*. PAMI, 30(2):283298, 2008.

[11] C.E. Rasmussen, C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.